UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/449,782	11/26/1999	JAMES MCKEETH	MICS:0194	6698

52142 7590 04/26/2010
FLETCHER YODER (MICRON TECHNOLOGY, INC.)
P.O. BOX 692289
HOUSTON, TX 77269-2289

EXAMINER	
BROPHY, MATTHEW J	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
04/26/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 09/449,782	**Applicant(s)** MCKEETH, JAMES
	Examiner MATTHEW J. BROPHY	**Art Unit** 2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 February 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,7,10-15,18-21 and 23-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,7,10-15,18-21 and 23-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This office action is in response amendment filed February 19, 2010.

2. Claims 1-5,7,10-15,18-21 and 23-28 are pending.

Response to Amendment

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1, 5, 10, 11, 15, 18, 21 and 23-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior Art (hereinafter AAPA) in view of "The Windows NT Command Shell" by Tim Hill (hereinafter Hill) (1998) in view of USPN 6,182,279 to Buxton and further in view of US Patent 6,681,265 Halva, (hereinafter Hlava.)

Regarding Claims 1, 15 and 21, AAPA teaches:

invoking, by an application, a call of a command line utility… wherein the command line utility is a utility executable from a command line prompt **(AAPA Page 1, 17-21"The conventional technique by which a user application obtains command line utility output is shown in FIG. 1. After a temporary text file is created (block 100), the command line utility whose output is desired is invoked via a standard interface**

(block 102).");

receiving output from the command line utility **(AAPA Page 1, Line 21-22 "Output from the command line utility is piped to the temporary file (block 104),")**;

storing the command line utility output …at a location identified by the identifier **(AAPA Page 1, Line 21-22 "Output from the command line utility is piped to the temporary file (block 104),")**

and retrieving, by the application, the command line utility output … at the location identified by the identifier. **(AAPA Page 1, Line 22-23 "from which the application extracts and processes the desired data (block 106).")**

AAPA does not explicitly teach:

the application providing an identifier in the call of the command line utility

however this limitation is taught by Hill:

(Hill Page 11, "The > redirection symbol redirects command output to the specified file. For example: C:\>dir >c:\dir.txt

This example creates a text file C:\DIR.TXT containing the output of the DIR command. The > symbol can be placed anywhere in the command, but is typically placed at the end of the command. A space is permitted between the > symbol and the file name. If the file specified by the redirection symbol already exists, any existing contents are deleted before the command is executed.")

However, it would have been obvious, to one of ordinary skill level in the art, at the time of the invention, to modify the method / system / storage device, as disclosed in AAPA, FIG. 1, using WindowsNT known redirection and piping commands, because piping the output of a script command to a file specified in the call of the command line utility allows the program to specify the exact system memory location in which the output will be stored. The combination is obvious because teachings are found in the prior art, and combined, with no change in functionality. It is a mere use of common sense by one skilled in the art to select and combine such known elements with no new function, i.e., a predictable result. The predictable result, utility output directly stored to a system storage, at a location identified by an identifier. A subsequently invoked extraction will retrieve the modified values from the temporary file.

AAPA does not teach:

storing …in a system registry database in a location identified by an identifier..

However, this limitation is taught by Hlvana. **(Col. 5, Ln 41-48, "FIG. 2 is a block diagram that conceptually illustrates interactions among entities according to one embodiment of the present invention. Command file 200 is prepared by a developer for the purpose of accessing a data store 240 such as the Windows Registry. This command file 200 executes the command file generator 210.The command file generator 210 provides access to a data store 240 such as the Windows Registry through a data store API 230 such as a Windows Registry API."** ***Storage is further suggested by the two-way arrow to registry 240 in FIG. 2,***

See Further "At step 340, the temporary command file 380 makes the configuration information [from registry locations] available through temporary environment variables, for example.") *[Inherently, the information stored in the registry data store must be at a location identified by an identifier, because in order to return the correct configuration information for each of the temporary variables, the system of Hlava must identify the location in the data store 240 to be accessed by API 230.]*

and retrieving, by the application…in a system registry database at the location identified by the identifier **(Col. 5, Ln 41-48, "FIG. 2 is a block diagram that conceptually illustrates interactions among entities according to one embodiment of the present invention. Command file 200 is prepared by a developer for the purpose of accessing a data store 240 such as the Windows Registry. This command file 200 executes the command file generator 210.The command file generator 210 provides access to a data store 240 such as the Windows Registry through a data store API 230 such as a Windows Registry API." See Further "At step 340, the temporary command file 380 makes the configuration information [from registry locations] available through temporary environment variables, for example.")**

It would be obvious to one of ordinary skill level in the art, at the time of the invention, to modify the method / system / storage device, as disclosed in AAPA, with the registry storage and retrieval taught by Hlvana because AAPA & Hill teach the desirability of storing output of command line utilities as described above, and Hlava teaches the desirability of providing registry access to command line operations (Col 2 Ln 37-46, “Advantageously, this method allows command files to access a data store such as the Windows Registry [which is increasingly popular in storing configuration information] without some of the problems associated with prior approaches…”). Therefore one of ordinary skill in the art would be motivated to modify the AAPA and Hill’s redirection to redirect to the Registry of Hlvana using the API 230 because it would allow storage of such output in the registry which “is increasingly used to store application configuration information.” (Col. 1, Ln 28-30.)

AAPA, Hill and Hlvana further teach limitations of dependent claims as described here below.

Regarding Claim 5, Hlvana teaches:

-system registry database comprises an operating system registry database.
(Col. 5, Ln 41-48, “FIG. 2 is a block diagram that conceptually illustrates interactions among entities according to one embodiment of the present invention. Command file 200 is prepared by a developer for the purpose of accessing a data store 240 such as the Windows Registry. This command file 200 executes the command file generator 210.The command file generator 210

provides access to a data store 240 such as the Windows Registry through a data store API 230 such as a Windows Registry API.")

Regarding Claim 10, Hill teaches:

-receiving output directly from the command line output utility.

(Hill Page 11, "The > redirection symbol redirects command output to the specified file. For example: C:\>dir >c:\dir.txt

This example creates a text file C:\DIR.TXT containing the output of the DIR command. The > symbol can be placed anywhere in the command, but is typically placed at the end of the command. A space is permitted between the > symbol and the file name. If the file specified by the redirection symbol already exists, any existing contents are deleted before the command is executed.")

.

Regarding claim 11, Hill teaches:

-receiving output from the command line output utility through a subsequent command line output routine.

(See Hill Page 11, e.g. Command redirection ">" & "cmd1 | cmd2")

Regarding claim 18, Hill teaches:

-instructions to receive one or more lines of output from the command line utility.

(Hill Page 11, "The > redirection symbol redirects command output to the specified file. For example: C:\>dir >c:\dir.txt
This example creates a text file C:\DIR.TXT containing the output of the DIR command. The > symbol can be placed anywhere in the command, but is typically placed at the end of the command. A space is permitted between the > symbol and the file name. If the file specified by the redirection symbol already exists, any existing contents are deleted before the command is executed.")

And Hlvana teaches:

-instructions to store each of said one or more lines of output in the <u>system registry database</u>

(Col. 5, Ln 41-48, "FIG. 2 is a block diagram that conceptually illustrates interactions among entities according to one embodiment of the present invention. Command file 200 is prepared by a developer for the purpose of accessing a data store 240 such as the Windows Registry. This command file 200 executes the command file generator 210.The command file generator 210 provides access to a data store 240 such as the Windows Registry through a data store API 230 such as a Windows Registry API." ***Storage is further suggested by the two-way arrow to registry 240 in FIG. 2*****, See Further "At step 340, the temporary command file 380 makes the configuration information [from registry locations] available through temporary environment variables, for example.")**

Regarding claim 23, Buxton teaches:

-the command line utility comprises a first command line utility, and wherein invoking the call by the application comprises invoking a call to pipe output of a second command line utility to the first command line utility…

-wherein storing the command line utility output comprises storing the command line utility output of the first command line utility.

(See Hill Page 11, e.g. Command redirection ">" & "cmd1 | cmd2")

Additionally see rejection of claim 1 above.

Regarding claim 24, Buxton teaches:

-the command line utility comprises a first command line utility, and wherein invoking the call by the application comprises invoking a call to pipe output of a second command line utility to the first command line utility…

-wherein storing the command line utility output comprises storing the command line utility output of the first command line utility.

This is a 'program storage device' version of claim 23 above. See rejection of claim limitations in claims 15 and 23 above.

Regarding claim 25, Buxton teaches:

-the command line utility comprises a first command line utility, the system further comprising a second command line utility, the application to invoke a call that causes output of the second command line utility to be piped to the first command line utility…

-the location identified by the identifier to store output of the first command line utility.

This is a 'system' version of claim 23 above. See rejection of claim limitations in claims 21 and 23 above.

Regarding Claims 26-28, Hill teaches: wherein receiving output from the command line utility comprises receiving output without creating [or using] a temporary file. **(Hill Page 11, "The > redirection symbol redirects command output to the specified file. For example: C:\>dir >c:\dir.txt This example creates a text file C:\DIR.TXT containing the output of the DIR command. The > symbol can be placed anywhere in the command, but is typically placed at the end of the command. A space is permitted between the > symbol and the file name. If the file specified by the redirection symbol already exists, any existing contents are deleted before the command is executed.")** *[Here, note that nowhere does Hill suggest that the DIR.txt must be temporary. Further, with respect to the Hlvana reference, while Hlvana describes using temporary variables, Hlvana contemplates the use of both temporary and non-temporary variables as evidenced by, e.g. Claim 2 of Hlvana, where information is stored in an "environment variable" as opposed to the "temporary environment variables" of claim 4 in Hlvana.]*

5. Claims 2-4,7, 12-14, 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant's Admitted Prior Art (hereinafter AAPA) in view of "The Windows NT Command Shell" by Tim Hill (hereinafter Hill) (1998) in view of US Patent 6,681,265 Halva, (hereinafter Hlava.) as applied above and further in view of USPN 6,182,279 to Buxton.

Regarding Claim 2, AAPA does not teach:
-providing the identifier comprises providing an identifier that identifies one or more entries in <u>the</u> system registry database.
However, this limitation is taught by Buxton:
(Fig. 2, item 205 and col. 13, lines 14-15, "…registry keys are created…" Also see col. 14, lines 29-59, "To facilitate loading of template onto another system…a number of registration key or subkey are included with template. Each template may have the keys 450A-I, as illustrated in Fig. 4C…Key 450H contains information indicating the name of the storage object in template storage file where initialization data…may be located…Key 450I contains information identifying the CLSID…)

In addition it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the command line redirection (to temporary files) of AAPA with the system register storage of Buxton as the use of the system registry provides the ability to fully use a system registry, avoid dual maintenance in temporary files and avoid inefficient use of programs as recognized by US patent 6,681,265 to Hlvana (Col 2 Ln 37-46, "Advantageously, this method allows command files to access a data store such as the Windows Registry without some of the problems associated with prior approaches. First, access to the data store can be achieved through a command file which is easier to write and maintain than a program. Secondly, there is no concern about synchronization between the data store and permanent environmental variables used to mimic the data store. Finally, the method allows full use of the data store as intended, that is, as a central repository for configuration type information.")

Regarding claim 3, Buxton teaches:

-providing a root key identifier.

(Col. 11, line 2: "Most OLE object application information is stored in subkeys under the CSLID root key…" Also see col. 17, lines 35-41, "Component loader loads, verifies and checks the license of a component by replacing in registry the InProcessServer 32 entry, i.e. key 450A…and adding additional registry keys 450B-J, as previously described, that will let the component loader (receiving a root key identifier) then load the correct OLE control.")

Regarding claim 4, Buxton teaches:

-providing a sub-key identifier.

(Col. 11, line 2 and col. 14, line 31: To facilitate loading of template…a number of registration or subkey are included with template…")

Regarding claim 7, Buxton teaches:

-providing the identifier comprises providing an identifier indicating the system registry database.

(Col. 10, line 66 – col. 11, line 4: A CLSID identifies the functionality of an object class that can display…access to property values…A subkey is used by an OLE to find out information about the control.")

Regarding claim 12, Buxton teaches:

-associating each line of command line utility output with a line identifier in the system registry database

As an example, (col. 3, lines 1-9) "Template storage with a means for indexing, including key information associated with the template. "…a memory having one or more locations, means for indexing one or more locations within the memory…" Also col. 13, lines 35-44, templates are stored with an enumerated decimal number: "Each template is stored in an ISTORAGE whose name is unique…and may have the form TEMPLEnnn, where nnn may be a decimal number.")

Regarding claim 13, Buxton teaches:

-setting each line identifier to a value corresponding to a position of that line in the command line utility output.
(Rejection of claim 12 is incorporated and further claim contains limitations as recited in claim 12. Therefore claim 13 is rejected under the same rational as claim 12.)

Regarding claim 14, Buxton teaches:
-setting a default value of the provided identifier to equal the total number of command utility output lines stored in the <u>system registry database</u>. (Rejection of claim 12 is incorporated and further claim contains limitations as recited in claim 12. Therefore claim 14 is rejected under the same rational as claim 12: As an example, (col. 3, lines 1-9) "Template storage with a means for indexing, including key information associated with the template. "…a memory having one or more locations, means for indexing one or more locations within the memory…" Also col. 13, lines 35-44, templates are stored with an enumerated decimal number: "Each template is stored in an ISTORAGE whose name is unique…and may have the form TEMPLEnnn, where nnn may be a decimal number.")

Regarding claim 19, Buxton teaches:
-instructions to associate a unique identifier with each of the one or more lines of output stored in the <u>system registry database</u>
See rejection of limitations in claim 2 above.

Regarding claim 20, Buxton teaches:

-instructions to set a value associated with the received identifier in the <u>system registry database</u> equal to the number of lines of output stored in the <u>system registry database.</u>

(Rejection of claim 18 is incorporated and further claim contains limitations as recited in claim 12. Therefore claim 20 is rejected under the same rational as claim 12.)

Response to Arguments

6. Applicant's arguments filed August 24, 2009 have been fully considered but they are not persuasive. Applicant's arguments with regards to the previous rejections of Claims 1, 15 and 21 are moot in view of the new grounds for rejection presented above.

In remarks, Applicant Argues:

> The
>
> Further, in the rejection, the Examiner cited the redirection symbol, ">", as disclosing an "identifier in the call of the command line utility." Office Action mailed November 19, 2009, pages 2-3. However, Applicant maintains that neither the redirection symbol (">") nor the specified file is a "call of the command line utility." As clearly stated in Hill, "[c]ommand redirection symbols are not visible to the command." Hill, page 10. Further, Hill states that "the shell processes them before the command is executed and they are not passed as arguments to the command." Id. Thus, the redirection symbol (">") is not any call of the command line utility. For example, the command line utility does not process the redirection symbol (">"). Indeed, as stated in Hill, the shell processes (e.g., executes) the redirection facility before the "command," e.g., command line utility, itself. Further, Applicant asserts that the "specified file" after the redirection symbol (">"), such as "dir.txt" described in Hill, is not an identifier in the call of the command line utility. The "specified file" is an argument of the redirection symbol. For example, as shown in table 2.4 of

> Hill, the "command redirection symbols" are provided as ">file" wherein "file" is the name of the specified file. Id. When using the redirection symbol, the command line utility does not receive the "specified file" as a call. Instead, the "specified file" is a part of and is used by the redirection symbol. Thus, Hill does not disclose "the application providing an identifier in the call of the command line utility" as recited in independent claims 1, 15, and 21.

Examiner's Response:

Examiner respectfully disagrees. Consistent with USPTO practice, In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., passing an identifier as an argument or other limiting interpretation of "in the call") are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Specifically, the claim includes the term "providing an identifier in the call of the command line utility" to which the examiner aligned, for example, the redirection of the dir command taught in Hill. Applicant's arguments with regards to ">" and/or the .txt file not being "in the call of the command line utility", examiner disagrees because the quoted example code line is a call of the Dir command, which includes the dir.txt identifier of a file used in Hill. While applicant points out that ">" and "dir.txt" are not passed as arguments to "dir" the argument is moot with respect to the claims because the claims only require the identifier (dir.txt) be "*in the call of the command line utility*", not proscribing any particular limitation on *how* the identifier appears or is used in the call, just that it is "in the call."

In remarks, Applicant Argues:

> Specifically, Applicant maintains that Buxton teaches away from such a combination with Hill and Halva. Hill is a reference directed to the "Windows NT Command Shell." Hill, page 1. Hill is directed to usage of the "command shell," a "command prompt," i.e., a command line, and various commands executed from the "command shell" by typing these commands into the "command prompt." Id. Similarly, Halva is directed to "command files" that are described therein as "a file containing one or more command line operations." Halva, col. 4, lines 10-20. Thus, both Hill and Halva are directed to usage of the "command line" and various commands executed from the command line. In contrast, Buxton discloses "OLE libraries" that are defined as "system-level services which utilize the interfaces defined by the COM specification" that call a "WIN 32 API." Buxton, col. 8, lines 6-8. Applicant asserts that there is a clear difference between a service and a command executed from the command prompt as recited in Hill, and between a service and a command line operation as recited in Halva. Further, as known to those of ordinary skill in the art and as stated in Buxton, API's are "application program interfaces" which are also quite different than a utility and a "command line utility." As they are described in Buxton, neither "application program interfaces" nor "system-level services" are "executable from a command line prompt," and thus cannot be considered a "command line utility." Applicant asserts one skilled in the art would not seek to combine Hill and Halva, directed to command line usage, with Buxton, directed to usage of system-level services, e.g., OLE libraries.

Examiner's Response:

Examiner respectfully disagrees. Applicant's continued discussion of Buxton's "OLE libraries" or "system-level services" is unpersuasive with respect to *both* the limitations of the invention AND the combination of the references, being that applicant admits that these arguments do not apply to the teachings of the limitations, and that these arguments have been address on the record already, the examiner now addresses the bearing of "OLE libraries" on the *combination*. As the examiner has quoted in the rejection, Hlvana teaches the advantageous nature of allowing command files to use the Windows registry. (Col. 2, Ln 37-46). Insomuch as both Hlvana and Buxton involve storage of data in the *registry* the would be obvious to combination to one of ordinary skill in the art at the time of the invention because of the advantages highlighted by Hlvana as well as the particular implementation details of that registry provided by Buxton.

In remarks, Applicant Argues:

> The
>
> Applicant respectfully disagrees with the Examiner' s interpretation. In particular, the Examiner is misinterpreting the environment variables of Hlava and the "variables described to temporary locations" of Hill. First, the portion of Hill cited by the Examiner does not describe or include any variables. Hill describes redirecting data to a file or to another command using the redirection and pipe commands. Hill does not discuss or mention redirecting output to variables. Hill, pages 10-11. Second, the "environment variables" of Hlava are a specific type of variables used to store application configuration information. Hlava, col. 1, lines 27-34. There is no discussion in Hill of any of these environment variables or how the commands disclosed therein interact with environment variables. Thus, to the extent that the Examiner's rational underpinning for the combination relies on the commonality between the "variables" of Hill and the "environment variables" described in Hlava, Applicant asserts there is no such commonality or basis for the combination, as Hill does not teach or suggest "variables," let alone the environment variables described in Hill. One of ordinary skill in the art would not seek to use the commands of Hill, such as redirection commands and pipe commands, with the system of Hlava that is clearly directed to the interaction between environment variables and the Windows Registry.

Examiner's Response:

Examiner respectfully disagrees. First, to clarify, it is believed where applicant wrote "as Hill does not teach or suggest "variables," let alone the environment variables described in Hill" that the applicant in fact intended "the environment variables described in *Hlvana*, and thus examiner now responds on that basis as the plain reading of the argument is illogical.

Examiner respectfully disagrees, furthermore, that the "variables" in Hlvana and the "file" of Hill cannot be combined. While these items are labeled differently, they both essentially comprise data storage of command output. (Compare Hill Page 11, DIR.txt used to store output, with e.g. FIG. 2 of Hlvana). As such, one of ordinary skill in the art would be motivated to uses these two different data storage types as a basis to combine the inventions, to allow the redirection in Hill to be done to a local storage, that might then be interchanged with the Windows Registry as contemplated by Hlvana. While they are different storage types, they are both storage, and are functionally related, and therefore the examiner respectfully disagrees with the Applicant's assertion that these elements are not combinable by one of ordinary skill in the art.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to MATTHEW J. BROPHY whose telephone number is 571-270-1642. The examiner can normally be reached on Monday-Thursday 8:00AM-5:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MJB

4/22/2010

/Anna Deng/
Primary Examiner, Art Unit 2191